UBRI-604 at AAAI-CAD21 Shared Task: Predicting Emphasis in Presentation Slides

GangQiang Hu[1], Chao Feng[1], HaoWen Lin[1], and JianGeng Chang[2]

[1]University of Science and Technology of China
[2]National University of Singapore
{hugq, chaofeng, linhaowen0421}@mail.ustc.edu.cn
{ephjia}@nus.edu.sg

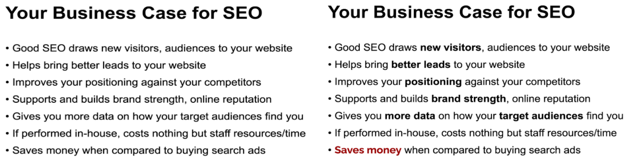
Abstract

This paper describes the system designed to addressing the problem proposed in AAAI-CAD21 shared task: Predicting Emphasis in Presentation Slides. We designed an end-to-end Transfomer-based system that takes the sentences in the slides as input and predicts the importance of the words in sentences. With the help of pre-trained language models, we only need to add some additional layers according to the downstream task and then finetune the whole model to achieve a good performance. ALBERT, GPT-2, ROBERTA, ERNIE 2.0, XLNET, XLM-ROBERTA and BERT are tried and we found that XLM-ROBERTA achieves the best average score. Additional lexical features of texts are also added to improve the performance of the system. Our system achieved the best $Match_m$ score of 0.525 and ranked first on the evaluation leaderboard [1],

Introduction

Emphasis Selection was first proposed by (Shirani et al. 2019) to choosing candidates for emphasis in visual media. With Emphasis Selection techniques, important candidates in visual media will be emphasized with different colors and fonts, which can attract the attention of readers and convey the right information to them effectively and without ambiguity. The purpose of this shared task is to design automatic methods for emphasis selection, i.e. choosing candidates for emphasis in presentation slide texts, to enable automated design assistance in authoring. Such as Figure 1, the words in the left slide have the same color and font, thus, the audience needs to read the slide from beginning to end with all their attention. However, the important words in the right slide are emphasized with different colors and fonts, which can guide the audience to focus on a few words and retain their attention to the speaker.

The task on predicting emphasis in presenting slides can be formulated to a sequence labeling problem. The major challenge of the task is that the dataset provided is collected from a variety of presentation slides with different topics, which means the distributions of the train dataset, development dataset and test dataset may be different. Besides, different annotators have different standards as their understanding to the texts may be different. Thus, these characteristics of the dataset requires out model to have a good generalization ability. We use a Transfomer-based (Vaswani et al. 2017) model as the pretraining model and a fully connected feedforward layer as a classifier model. Different bert-based language model, such as BERT (Devlin et al. 2018), ROBERTA (Liu et al. 2019), GPT-2 (Radford et al. 2019), ERNIE 2.0 (Sun et al. 2020), XLNET (Yang et al. 2019) and ALBERT (Lample and Conneau 2019), are tried. These unsupervised language models are pre-trained on a large amount of unlabeled data and carry lots of lexical, syntactic and semantic information. They will provide rich text information in the open domain and remarkably improve the generalization ability of the model. With these pre-trained language model, we only need to add an additional layer on them and finetune the model on downstream tasks. Finally, we apply feature engineering and leverage the lexical features of the dataset for further improvement.



Your Business Case for SEO

- Good SEO draws new visitors, audiences to your website
- Helps bring better leads to your website
- Improves your positioning against your competitors
- Supports and builds brand strength, online reputation
- Gives you more data on how your target audiences find you
- If performed in-house, costs nothing but staff resources/time
- Saves money when compared to buying search ads

Your Business Case for SEO

- Good SEO draws **new visitors**, audiences to your website
- Helps bring **better leads** to your website
- Improves your **positioning** against your competitors
- Supports and builds **brand strength**, online reputation
- Gives you **more data** on how your **target audiences** find you
- If performed in-house, costs nothing but staff resources/time
- **Saves money** when compared to buying search ads

Figure 1: The left slide is plain and harder for audience to process. The important words in the right slide are emphasized with different color and font

The rest of the paper is organized as follows. In the Background section, we analyze the dataset and describes the evaluation metric. In the System Overview section, we describe the system we proposed. In the Experiment section, we analyze the results of the experiments.

[1] https://github.com/Daemon-ser/EmphasisSelection.git

Background

Data Description

This dataset is collected from a variety of presentation slides with different topics. Each instance represents one slide page along with annotations. The data is randomly devided into training (70%), development (10%), and test (20%) sets for further analysis. The detail analysis of the dataset is shown in Table 1.

Table 1: The detail statistic of annotated dataset

	Slides	Sentences	Words
Train	1241	8868	96943
Dev	180	1177	12822
Test	355	2571	28108
Total	1776	12616	137873

Evaluation Metric

$Match_m$: For each instance X in the test set D_{test}, we select a set $S_m^{(x)}$ of $m \in (1, 5, 10)$ words with the top m probabilities according to the ground truth. Analogously, we select a prediction set set $\widehat{S}_m^{(x)}$ for each m, based on the predicted probabilities.

The metric $Match_m$ is defined as follows:

$$Match_m := \frac{\sum_{x \in D_{test}} |S_m(x) \cap \widehat{S}_m^{(x)}|/|x|}{|D_{test}|}$$

Related Work

Recent years have witnessed the rapid development of language models. It is shown that natural language process tasks, such as named entity recognition and reading comprehension, can achieve better performance by pre-trained language models.

BERT, as a milestone of the development of language models, starts a new era in natural language processing. BERT can be pre-trained unsupervisely on a large amount of unlabeled data by several strategies such as masked language and next sentence prediction. After the release of BERT, a number of transformer-based language models are proposed. XLNet combines the advantages of autoregressive and autoencoding and proposed a generalized autoregressive language model. ERNIE 2.0 further considers the word, structure and semantic level information in training corpora by incrementally builds pre-training tasks and then learn pre-trained models on these constructed tasks. ROBERTA proposed new methods to training BERT and achieve better performance. ALBERT is a lite version of BERT which decreases the number of parameters and retains the performance of BERT.

Shirani et al. introduces some SOTA models (Shirani et al. 2020) in SeEval-2020 Task 10: Emphasis Selection for Written Text in Visual Media. See (Shirani et al. 2021) for a further description of the task and data.

System Overview

Transfomer-based Approach

As illustrated in Figure 2, we use bert-based pretraining models with a fully connected layer. The inputs of our system are sentences and the outputs are the score of sentences. At first, the tokenizer of the corresponding bert-based pretraining model converts the sentences to token ids. Since the length of the sentence and the length of tokens are different, the start indexes of each token are also recorded for later computation of the final score. The token ids are the input of bert-based pretraining model. We use the pre-trained models in huggingface-transfomers and fine-tine them in the dataset that the task provided. To better extract the semantic information of the sentence in different levels, all layers of the pretraining model are concatenated.

Also, we observed that the word start with capital letter often gets a higher score and the word containing special characters (such as .,:) often get a lower score. These lexical features are also concatenated with the output of the pretraining model, and finally, they are fed into a three-layer feedforward network with dropout in the input of each layer except the last layer. The outputs of the feedforward network are scaled to (0,1) by a sigmoid function. Finally, the scores of each word are computed based on the score of the first token of the word.

Feature Engineering

After further observing and analyze the dataset, we find that adding some lexical features of words can bring further improvements. As the statistic results in Table**??** shown, the word start with capital letter often get a higher score and the word containing special characters (such as .,:) often get a lower score. The special characters here are the characters except for letters and numbers.

The lexical features are designed and numeralized as follows:

$$feature\ 1 = \begin{cases} 0, & every\ character\ is\ special \\ 1, & some\ characters\ are\ special \\ 2, & no\ character\ is\ special \end{cases}$$

$$feature\ 2 = \begin{cases} 0, & word\ in\ uppercase \\ 1, & word\ starts\ with\ capital\ lettter \\ 2, & other \end{cases}$$

The two features are finally concatenated with the output of transfomer-based models and then fed into the feedforward layer.

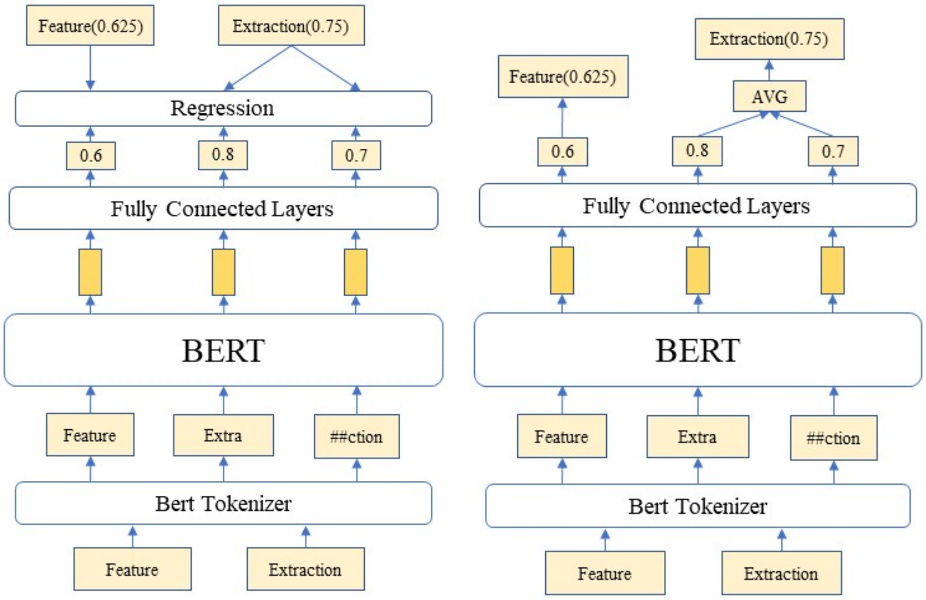
Experiment

Setup

We use PyTorch[2] library as the deep learning framework and huggingface-transfomers[3] as the pre-trained language model framework. All experiments are executed in an Nvidia V100 SXM2 GPU. Each model runs 20 epochs with 5 early stop count based on the average $Match_m$ score.Since the distributions of train dataset, development dataset and test dataset

[2] https://pytorch.org/

[3] https://huggingface.co/transformers/



(a) Training

(b) Inferencing

Figure 2: Figure 2(a) is the training process of the model. The label of the token is assigned as the label of the word it belongs to. Figure 2(b) is the inference process of the model. The predicted score of the word is the average of the scores of tokens belongs to the corresponding word.

Table 2: Average gain of models

Model Name	Ave. Gain(+ Feat 1)	Avg. Gain(+ Feat 2)	Avg. Gain(+ both)
ALBERT	0.0010	0.0004	0.0011
GPT-2 LARGE	0.0007	0.0007	0.0009
ERNIE 2.0	0.0011	0.0009	0.0012
ROBERTA	0.0013	0.0010	0.0015
XLM-RoBERTa-LARGE	0.0012	0.0009	0.0015
BERT-LARGE-CASED	0.0013	0.0008	0.0014
XLNET-LARGE-CASED	0.0013	0.0009	0.0014

are different, simply learning from the train dataset and validating in the development dataset will decrease the generalization ability of the model. We combine all the provided training and development datasets, then split them under a random 5-fold setting.

Each layer of the pre-trained language models is not frozen. We choose a three-layer feedforward network with dropout mechanism before each layer except the output layer. The hidden sizes of the three-layer feedforward layers are 100, 40, 1. The dropout rate is set to 0.3. We use AdamWoptimizer. We use Binary Cross-Entropy loss to train the model and $Match_m$ score is used as the performance metric. The learning rate is set to 1e-6 and the batch size is set to 16. To make sure the experiment result is reliable and comparable among different models, we fixed the random seed of the system. In addition, the random seed of the function that dividing the dataset into 5 fold is fixed to make sure the input data of each training have no differences.

Table 3: Average score of models

Model Name	Ave. Score
ALBERT	0.8513
GPT-2 LARGE	0.8397
ERNIE 2.0	0.8537
ROBERTA	0.8598
XLM-RoBERTa-LARGE	**0.8632**
BERT-LARGE-CASED	0.8578
XLNET-LARGE-CASED	0.8595

Results

We tried different pre-trained language model and do ablation studies on the contributions of each additional lexical feature.

As Table 3 shown, XLM-ROBERTA-LARGE achieve the best average score as the ability to extract information from texts of XLM-ROBERTA is superior to BERT. GPT-2-LARGE achieves the lowest average score as the ability to extract information from texts of GPT-2 is inferior to BERT. In addition, the performance of ALBERT is poor as it is a tiny version of BERT and has a smaller number of parameters than BERT.

As Table 2, both lexical feature 1 and lexical feature 2 have contributions to the improvement of the performance of the model. Feature 1 makes more improvements than feature 2, which means that the words containing special letters are often not important but the words start with capital letter sometimes is not important too.

Conclusion

We describes the system designed to addressing the problem proposed in CAD21@AAAI21 shared task: Predicting Emphasis in Presentation Slides. We designed an end-to-end Transfomer-based system. And we tried ALBERT, GPT-2, ROBERTA, ERNIE 2.0, XLNET, XLM-ROBERTA and BERT as pre-trained language models. Experiment results show that XLM-ROBERT achieves the best average score.

References

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Lample, G., and Conneau, A. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8):9.

Shirani, A.; Dernoncourt, F.; Asente, P.; Lipka, N.; Kim, S.; Echevarria, J.; and Solorio, T. 2019. Learning emphasis selection for written text in visual media from crowd-sourced label distributions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1167–1172.

Shirani, A.; Dernoncourt, F.; Lipka, N.; Asente, P.; Echevarria, J.; and Solorio, T. 2020. Semeval-2020 task 10: Emphasis selection for written text in visual media. *arXiv preprint arXiv:2008.03274*.

Shirani, A.; Tran, G.; Trinh, H.; Dernoncourt, F.; Lipka, N.; Asente, P.; Echevarria, J.; and Solorio, T. 2021. Learning to emphasize: Dataset and shared task models for selecting emphasis in presentation slides. In *Proceedings of CAD21 workshop at the Thirty-fifth AAAI Conference on Artificial Intelligence (AAAI-21)*.

Sun, Y.; Wang, S.; Li, Y.-K.; Feng, S.; Tian, H.; Wu, H.; and Wang, H. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *AAAI*, 8968–8975.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems* 30:5998–6008.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, 5753–5763.